Customer Purchase Pattern Analysis pandas data cleaning

January 7, 2026

[11]:
```
# Loading the dataset using pandas
```

[12]:
```
import pandas as pd

path = r"C:\Users\shrav\OneDrive\Desktop\SHRIDHAR\CAPSTONE␣
 ↪project\customer_shopping_behavior.csv"

df = pd.read_csv(path)
print(df.head())
```

```
   Customer ID  Age Gender Item Purchased  Category  Purchase Amount (USD)  \
0            1   55   Male         Blouse  Clothing                     53
1            2   19   Male        Sweater  Clothing                     64
2            3   50   Male          Jeans  Clothing                     73
3            4   21   Male        Sandals  Footwear                     90
4            5   45   Male         Blouse  Clothing                     49

        Location Size      Color  Season  Review Rating Subscription Status  \
0       Kentucky    L       Gray  Winter            3.1                  Yes
1          Maine    L     Maroon  Winter            3.1                  Yes
2  Massachusetts    S     Maroon  Spring            3.1                  Yes
3   Rhode Island    M     Maroon  Spring            3.5                  Yes
4         Oregon    M  Turquoise  Spring            2.7                  Yes

   Shipping Type Discount Applied Promo Code Used  Previous Purchases  \
0        Express              Yes             Yes                  14
1        Express              Yes             Yes                   2
2  Free Shipping              Yes             Yes                  23
3   Next Day Air              Yes             Yes                  49
4  Free Shipping              Yes             Yes                  31

  Payment Method Frequency of Purchases
0          Venmo            Fortnightly
1           Cash            Fortnightly
2    Credit Card                 Weekly
3         PayPal                 Weekly
4         PayPal               Annually
```

[13]:
```
# obtaining data information using df.head() command
```

[14]:
```
df.head()
```

[14]:

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season	Review Rating	Subscription Status	Shipping Type	Discount Applied	Promo Code Used	Previous Purchases	Payment Method	Frequency of Purchases
0	1	55	Male	Blouse	Clothing	53	Kentucky	L	Gray	Winter	3.1	Yes	Express	Yes	Yes	14	Venmo	Fortnightly
1	2	19	Male	Sweater	Clothing	64	Maine	L	Maroon	Winter	3.1	Yes	Express	Yes	Yes	2	Cash	Fortnightly
2	3	50	Male	Jeans	Clothing	73	Massachusetts	S	Maroon	Spring	3.1	Yes	Free Shipping	Yes	Yes	23	Credit Card	Weekly
3	4	21	Male	Sandals	Footwear	90	Rhode Island	M	Maroon	Spring	3.5	Yes	Next Day Air	Yes	Yes	49	PayPal	Weekly
4	5	45	Male	Blouse	Clothing	49	Oregon	M	Turquoise	Spring	2.7	Yes	Free Shipping	Yes	Yes	31	PayPal	Annually

[15]:
```
# taking info about data using .info() command
```

[16]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3900 entries, 0 to 3899
Data columns (total 18 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Customer ID             3900 non-null   int64
 1   Age                     3900 non-null   int64
 2   Gender                  3900 non-null   object
 3   Item Purchased          3900 non-null   object
 4   Category                3900 non-null   object
 5   Purchase Amount (USD)   3900 non-null   int64
 6   Location                3900 non-null   object
```

```
 7   Size                    3900 non-null   object
 8   Color                   3900 non-null   object
 9   Season                  3900 non-null   object
 10  Review Rating           3863 non-null   float64
 11  Subscription Status     3900 non-null   object
 12  Shipping Type           3900 non-null   object
 13  Discount Applied        3900 non-null   object
 14  Promo Code Used         3900 non-null   object
 15  Previous Purchases      3900 non-null   int64
 16  Payment Method          3900 non-null   object
 17  Frequency of Purchases  3900 non-null   object
dtypes: float64(1), int64(4), object(13)
memory usage: 548.6+ KB
```

[17]:
```
# Summary statistical data using .describe() with include='all' because we get␣
↪info about numerical as well as catagorical data also
```

[18]:
```
df.describe(include='all')
```

[18]:

	Customer ID	Age	Gender	Item Purchased	Category	\
count	3900.000000	3900.000000	3900	3900	3900	
unique	NaN	NaN	2	25	4	
top	NaN	NaN	Male	Blouse	Clothing	
freq	NaN	NaN	2652	171	1737	
mean	1950.500000	44.068462	NaN	NaN	NaN	
std	1125.977353	15.207589	NaN	NaN	NaN	
min	1.000000	18.000000	NaN	NaN	NaN	
25%	975.750000	31.000000	NaN	NaN	NaN	
50%	1950.500000	44.000000	NaN	NaN	NaN	
75%	2925.250000	57.000000	NaN	NaN	NaN	
max	3900.000000	70.000000	NaN	NaN	NaN	

	Purchase Amount (USD)	Location	Size	Color	Season	Review Rating	\
count	3900.000000	3900	3900	3900	3900	3863.000000	
unique	NaN	50	4	25	4	NaN	
top	NaN	Montana	M	Olive	Spring	NaN	
freq	NaN	96	1755	177	999	NaN	
mean	59.764359	NaN	NaN	NaN	NaN	3.750065	
std	23.685392	NaN	NaN	NaN	NaN	0.716983	
min	20.000000	NaN	NaN	NaN	NaN	2.500000	
25%	39.000000	NaN	NaN	NaN	NaN	3.100000	
50%	60.000000	NaN	NaN	NaN	NaN	3.800000	
75%	81.000000	NaN	NaN	NaN	NaN	4.400000	
max	100.000000	NaN	NaN	NaN	NaN	5.000000	

	Subscription Status	Shipping Type	Discount Applied	Promo Code Used	\
count	3900	3900	3900	3900	

unique	2	6	2	2
top	No	Free Shipping	No	No
freq	2847	675	2223	2223
mean	NaN	NaN	NaN	NaN
std	NaN	NaN	NaN	NaN
min	NaN	NaN	NaN	NaN
25%	NaN	NaN	NaN	NaN
50%	NaN	NaN	NaN	NaN
75%	NaN	NaN	NaN	NaN
max	NaN	NaN	NaN	NaN

	Previous Purchases	Payment Method	Frequency of Purchases
count	3900.000000	3900	3900
unique	NaN	6	7
top	NaN	PayPal	Every 3 Months
freq	NaN	677	584
mean	25.351538	NaN	NaN
std	14.447125	NaN	NaN
min	1.000000	NaN	NaN
25%	13.000000	NaN	NaN
50%	25.000000	NaN	NaN
75%	38.000000	NaN	NaN
max	50.000000	NaN	NaN

[19]:
```
# Checking if missing data or null values are present in the dataset using␣
↪isnull() with .sum()
```

[20]:
```
df.isnull().sum()
```

[20]:
```
Customer ID                0
Age                        0
Gender                     0
Item Purchased             0
Category                   0
Purchase Amount (USD)      0
Location                   0
Size                       0
Color                      0
Season                     0
Review Rating             37
Subscription Status        0
Shipping Type              0
Discount Applied           0
Promo Code Used            0
Previous Purchases         0
Payment Method             0
Frequency of Purchases     0
```

```
dtype: int64
```

[21]:
```
# Imputing missing values in Review Rating column with the median rating of the product category
```

[22]:
```
df['Review Rating'] = df.groupby('Category')['Review Rating'].transform(lambda x: x.fillna(x.median()))
```

[23]:
```
# after replacing missing values with median values accordingly again verify is there any null value present in data
```

[24]:
```
df.isnull().sum()
```

[24]:
```
Customer ID               0
Age                       0
Gender                    0
Item Purchased            0
Category                  0
Purchase Amount (USD)     0
Location                  0
Size                      0
Color                     0
Season                    0
Review Rating             0
Subscription Status       0
Shipping Type             0
Discount Applied          0
Promo Code Used           0
Previous Purchases        0
Payment Method            0
Frequency of Purchases    0
dtype: int64
```

[25]:
```
# Renaming columns accordingly for better readability and documentation
```

[26]:
```
df.columns = df.columns.str.lower()
df.columns = df.columns.str.replace(' ','_')
```

[27]:
```
# after renaming column verify column data
```

[28]:
```
df.columns
```

[28]:
```
Index(['customer_id', 'age', 'gender', 'item_purchased', 'category',
       'purchase_amount_(usd)', 'location', 'size', 'color', 'season',
       'review_rating', 'subscription_status', 'shipping_type',
       'discount_applied', 'promo_code_used', 'previous_purchases',
       'payment_method', 'frequency_of_purchases'],
      dtype='object')
```

[ ]:
```
# rename column name 'purchase_amount_(usd)' to 'purchase_amount' for better readability
```

[ ]:
```
df.rename( columns={'purchase_amount_(usd)': 'purchase_amount'}, inplace=True)
```

[ ]:
```
# create a new column age_group according age range
```

[30]:
```
labels = ['Young Adult', 'Adult', 'Middle-aged', 'Senior']
df['age_group'] = pd.qcut(df['age'], q=4, labels = labels)
```

[ ]:
```
# viewing first 10 rows data after grouping
```

[31]:
```
df[['age','age_group']].head(10)
```

[31]:

	age	age_group
0	55	Middle-aged
1	19	Young Adult
2	50	Middle-aged
3	21	Young Adult
4	45	Middle-aged
5	46	Middle-aged
6	63	Senior
7	27	Young Adult
8	26	Young Adult
9	57	Middle-aged

[ ]:
```
# create new column purchase_frequency_days using map command
```

[32]:
```
frequency_mapping = {
    'Fortnightly': 14,
    'Weekly': 7,
    'Monthly': 30,
    'Quarterly': 120,
    'Bi-Weekly': 14,
    'Annually': 365,
    'Every 3 Months': 90
}

df['purchase_frequency_days'] = df['frequency_of_purchases'].map(frequency_mapping)
```

[ ]:
```
# viewing first 10 rows data after mapping
```

[33]:
```
df[['purchase_frequency_days','frequency_of_purchases']].head(10)
```

[33]:

	purchase_frequency_days	frequency_of_purchases
0	14	Fortnightly
1	14	Fortnightly

```
2                          7               Weekly
3                          7               Weekly
4                        365              Annually
5                          7               Weekly
6                        120            Quarterly
7                          7               Weekly
8                        365              Annually
9                        120            Quarterly
```

[ ]:
```
# verifying the data in column of 'discount_applied' and 'promo_code_used' with␣
↪fist 10 rows
```

[34]:
```
df[['discount_applied','promo_code_used']].head(10)
```

[34]:
```
  discount_applied promo_code_used
0              Yes             Yes
1              Yes             Yes
2              Yes             Yes
3              Yes             Yes
4              Yes             Yes
5              Yes             Yes
6              Yes             Yes
7              Yes             Yes
8              Yes             Yes
9              Yes             Yes
```

[ ]:
```
# verifying the data in column of 'discount_applied' and 'promo_code_used' with␣
↪all whole two column
```

[35]:
```
(df['discount_applied'] == df['promo_code_used']).all()
```

[35]:
```
np.True_
```

[ ]:
```
# after found both column data was same then one column will be deleted using␣
↪drop command
```

[36]:
```
df = df.drop('promo_code_used', axis=1)
```

[ ]:
```
# finally take look on all columns and complete data cleanining
```

[37]:
```
df.columns
```

[37]:
```
Index(['customer_id', 'age', 'gender', 'item_purchased', 'category',
       'purchase_amount', 'location', 'size', 'color', 'season',
       'review_rating', 'subscription_status', 'shipping_type',
       'discount_applied', 'previous_purchases', 'payment_method',
       'frequency_of_purchases', 'age_group', 'purchase_frequency_days'],
      dtype='object')
```

[ ]:
```
# connecting cleaned data to postgre SQL server for dta analysis
```

0.0.1 Connecting Python script to PostgreSQL

[ ]:
```
# installing necessary files for connecting data to server
```

[25]:
```
!pip install psycopg2-binary sqlalchemy
```

```
Requirement already satisfied: psycopg2-binary in
c:\users\shrav\anaconda3\lib\site-packages (2.9.11)
Requirement already satisfied: sqlalchemy in c:\users\shrav\anaconda3\lib\site-
packages (2.0.39)
Requirement already satisfied: greenlet!=0.4.17 in
c:\users\shrav\anaconda3\lib\site-packages (from sqlalchemy) (3.1.1)
Requirement already satisfied: typing-extensions>=4.6.0 in
c:\users\shrav\anaconda3\lib\site-packages (from sqlalchemy) (4.12.2)
```

[ ]:
```
# write command for data loading to postgre SQL server
```

[27]:
```
from sqlalchemy import create_engine

# Step 1: Connect to PostgreSQL
# Replace placeholders with your actual details
username = "postgres"      # default user
password = "root" # the password you set during installation
host = "localhost"         # if running locally
port = "5432"              # default PostgreSQL port
database = "customer_behaviour"    # the database you created in pgAdmin

engine = create_engine(f"postgresql+psycopg2://{username}:{password}@{host}:
 ↪{port}/{database}")

# Step 2: Load DataFrame into PostgreSQL
table_name = "customer"   # choose any table name
df.to_sql(table_name, engine, if_exists="replace", index=False)

print(f"Data successfully loaded into table '{table_name}' in database␣
 ↪'{database}'.")
```

```
Data successfully loaded into table 'customer' in database 'customer_behaviour'.
```

[ ]:

[ ]:

[ ]: